So google.com finally challenges the flash? I did not see this one coming!

Some people argue that the flash is by far the fastest man ever to exists since he is connected to the speed force. However, I think google.com is faster. In fact, google doesn’t need speed force to be fast because Google.com owns the flash.

Anyway, jokes aside, have you ever been curious what happens when you enter google.com on the address bar and press enter? They say computers understand machine language, 1’s and 0’s, and yet you may wonder, how does my computer understand English? Maybe it is special…. or not?

In this article, I will explain to you step by step what happens behind the scenes of typing google.com and pressing enter. I would however like to mention that the explanation may contain words that are not basic English. Not to worry, I have a well written thesaurus immediately after the explanation.

**So, what happens behind the scenes of google.com?**

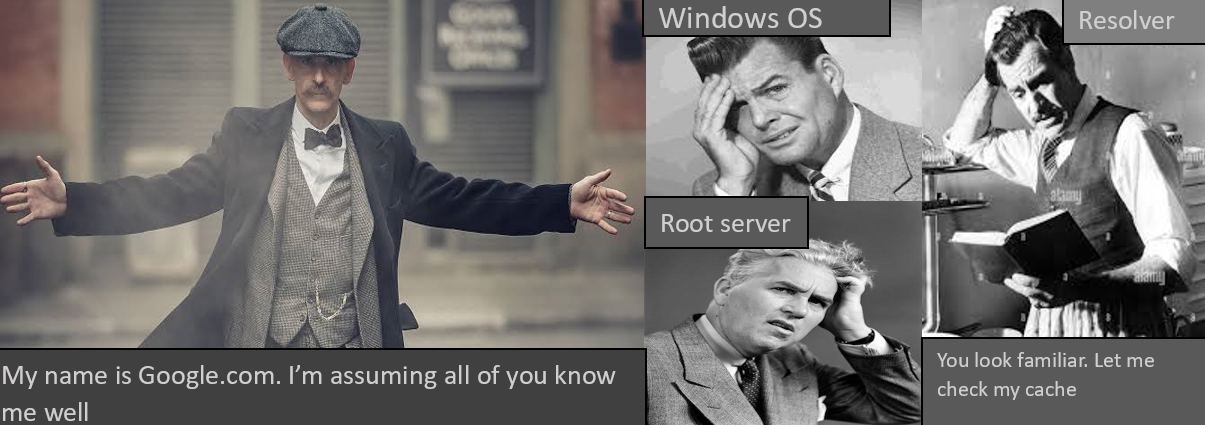
On opening your browser, google to be specific, you are met with the words “Search Google or enter URL”. In this case, you type "google.com" in your web browser's address bar and hit "Enter". All you see is a ‘line’ moving to the right and boom, you get what you wanted.

On hitting enter, Immediately, a request is sent to the Google **server** via the HTTP protocol. The browser attempts to check if it has any information stored about the website, but it does not recognize the URL. Here is why:

Computers connect to the internet using **IP addresses** to identify the location of the device they need to communicate to. The **domain** names like google.com are designed to be easily remembered by people, as they tend to find names more memorable than numerical addresses. However, computer systems use numerical addresses called IP addresses to identify websites on the internet.

When the request fails, the operating system (Linux or Windows) is requested to find google.com. However, the OS does not recognize this URL or its IP. However, the OS calls the resolver. The resolver is your internet service provider. The resolver checks its cache for this information.

 The cache is a high-speed data storage layer which stores a subset of data so that future requests for that data can be served up faster.



The resolver also fails to locate the IP address. It therefore has no choice but to ask the **root server.**

This is where we involve the **DNS.**DNS It is a technology that translates human-adapted text-based domain names to machine-adapted, numerical-based IPs.

The root server sits on top of the DNS hierarchy and is named (letter). root.server.net. The root server doesn’t know the URL or the IP address but it knows where to locate the .com **TLD server** which belongs to ICANN.

The resolver stores the information it’s fed by the root so that it doesn’t ask the root again. Mr. Root server goes to the .com TLD server. Unfortunately, it can’t provide the IP address of google.com but it knows its authoritative name servers. The .com TLD server points Resolver to the name servers. Why?

When a domain is purchased, the domain registrar reserves the name and it communicates to the TLD registry and the authoritative name servers. Authoritative name servers are like ns1.google.com.

To know who are the authoritative name servers for your domain, run a **WHOIS** query.

Finally, the authoritative name server provides the IP address and the resolver saves this information. The resolver gives the OS the IP address. What’s astonishing is that the above process happens within a second.

 Well, after obtaining the IP address, our device (the client) will establish a TCP connection with the server. Then the browser will compose an HTTP request and that request will be sent to the server. This request will contain a request line, request headers and an optional request message body. HTTP requests are not only used to fetch web pages. The request line contains a request method and by using different methods (GET, HEAD, PUT, POST, DELETE, …) we can do a variety of things like post data on the server, ask the server to store data, ask the server to delete data and so on…To fetch a web page usually a GET or HEAD method is used.

The server will process this request and send an HTTP response. This response will also include a response (status) line, headers and an optional body. The response line contains a status code reflecting the outcome of the request. Common status codes to know are:

1.      200 OK: request is fulfilled.

2.      301 Move Permanently: the requested resources have been moved permanently to a new location.

3.      302 Found and Redirect: the requested resources have been moved to a new location temporarily.

4.      400 Bad Request: the server could not interpret or understand the request, 401 Authentication Required: The requested resource is protected, and requires the client’s credential (username/password).

5.      403 Forbidden: Server refuses to supply the resource, regardless of the identity of the client.

6.      404 Not Found: The requested resource cannot be found on the server.

7.      500 Internal Server Error: The server is confused, often caused by an error in the server-side program responding to the request.

8.      501 Method Not Implemented: The request method used is invalid (could be caused by a typing error, e.g., “GET” misspell as “Get”).

**HOW DOES GOOGLE.COM REMEMBER WHAT YOU DID LAST TIME??**



Entering "google.com" in the web browser address bar is typically sufficient to access the site. If you have visited the website before, the browser will remember the port and protocol used in your previous interactions and apply them automatically. However, if this is your first time visiting the site, the browser will initiate an HTTP request, which will then be redirected to the more secure HTTPS protocol.

**DO BROWSERS HIRE NEW SECURITY? I SAW A GUY NAMED HSTS**

Most modern browsers support a security policy called HTTP Strict Transport Security (HSTS), which maintains a list of sites that must be accessed via HTTPS, even if a user types an HTTP URL. This list is called the Preload List and it helps prevent man-in-the-middle attacks and other security threats.

**DEFINITIONS**

1.      URL (A Uniform Resource Locator)- It is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. Examples: <https://www.google.com/>

2.      Firewall - Firewalls are hardware or software security implementations that filter out incoming and outgoing network traffic and are usually put between a private network and the Internet.

3.       TCP/IP (The Internet protocol suite)-A framework for organizing the set of communication protocols used in the Internet and similar computer networks according to functional criteria

4.      HTTPS - The “https” portion refers to the protocol and must be followed by a semi-colon and two forward slashes. The Hypertext Transfer Protocol (HTTP) is one of the major protocols in the TCP/IP suite. It is a stateless client-server protocol used to fetch web pages from the internet. HTTPS is a secure version of HTTP which uses the Secure Socket Layer (SSL) protocol for encrypting communication between the client and the server and securing user authentication.

5.      Resolver - This is your ISP (internet service provider)

6.      DNS- It is a technology that translates human-adapted text-based domain names to machine-adapted, numerical-based IPs.

7.      IP address – A unique address that identifies a device on the internet or local network.

8.      The “google.com” part is known as the domain name. A domain name is the website address. This is what users type in a browser search bar to directly access your website.

9.        The “.com” is called the top-level domain. TLD is everything that follows the final dot of a domain name. For example, in the domain name 'google.com', '.com' is the TLD. The root server normally knows where to locate the .com TLD server. Other types of top-level domains are .ORG and .NET

10.   The “Google” part is the second-level domain. A Second Level Domain (SLD) is part of the domain name that is located right before a Top-Level Domain (TLD)

11.   The “www” part is known as the subdomain. Subdomains are part of a domain that comes before the main domain name and domain extension

12.  The number at the end which is separated from the domain name by a colon is the port number. Ports are communication end-points identifying a specific process or type of network service enabling the use of multiple services from a single IP address. 443 is the port number associated with the HTTPS protocol.

**SERVERS**

In computing, a server is a piece of computer hardware or software that provides functionality for other programs or devices, called "clients." This architecture is called the client–server model.

However, it is important to note that any computer, including a personal laptop, can serve as a server if the necessary server software is installed on it. Some common types of servers include web servers, application servers, file servers, print servers, and proxy servers.

A web server – It is a software application that is specifically designed to handle HTTP requests and serve static web pages. These pages are typically written in programming languages such as HTML, CSS, and JavaScript. There are several commonly used web server software programs available, including Nginx, Apache, and Microsoft IIS.

Application servers- An application server is a type of server software that provides an environment for running and executing applications. It is designed to handle the processing and management of dynamic content, such as web-based applications that require server-side processing.

Database- a database refers to a collection of data that is organized and stored electronically. This data can be accessed and managed through software applications, via the Internet or through a local network. Websites often use databases to store and manage information, such as user account details, product inventory, and customer orders. Common types of databases used in web applications include relational databases, NoSQL databases, and cloud-based databases.

**Load distribution**

Load distribution- A standard way of distributing the load over an array of servers to return the same preference number for each server set. Load balancers distribute the workload of a system to increase reliability, efficiency and availability. These can be hardware or software.

Software load balancers implement one or more scheduling algorithms to achieve their purpose. The most common algorithms include:

1.     Round Robin - Requests are distributed evenly across servers in a sequential manner.

2.     Least Connection - Requests are routed to the server with the least active connections.

3.     IP Hash - The server selection is based on the hash value of the client's IP address.

4.     Random - Requests are randomly assigned to servers.

5.     Weighted Round Robin - The same as Round Robin, but with different weights assigned to servers to reflect their processing power.

6.     Weighted Least Connection - The same as Least Connection, but with different weights assigned to servers based on their processing capacity.

**LOAD BALANCING ALGORITHM BASED ON LAYER CONTEXT**

1.     Layer 4 load balancing: This type of load balancing operates at the transport layer of the network stack, using information such as IP addresses and port numbers to distribute traffic.

2.     Layer 7 load balancing: This type of load balancing operates at the application layer of the network stack, using information such as HTTP headers, cookies, and URLs to distribute traffic.

3.     Layer 5 load balancing: This type of load balancing operates at the session layer of the network stack, using information such as session IDs to distribute traffic.

The choice of layer context depends on the specific needs and requirements of the web application, such as the type of traffic being handled and the level of control over traffic distribution.

Well, hope you learnt something.